

Phân Tích Dữ Liệu
Thực Tế với Python
**Bài 7.1: Làm Quen
Thư Viện Pandas**

PYTHON
PANDAS
DATA SCIENCE



Quang-Khai Tran, Ph.D
CyberLab, 03/2023

(Ảnh: Internet)



Nội dung

1. Làm quen với Pandas
2. Ghi/đọc file với Pandas
3. Vẽ biểu đồ với Pandas
4. Thảo Luận





Phần 1.

Làm quen với Pandas



1.1. Giới thiệu Pandas

1.2. Series

1.3. DataFrame

1.4. Sự tương quan

- Giữa các cột trong DataFrame
- Giữa các DataFrame

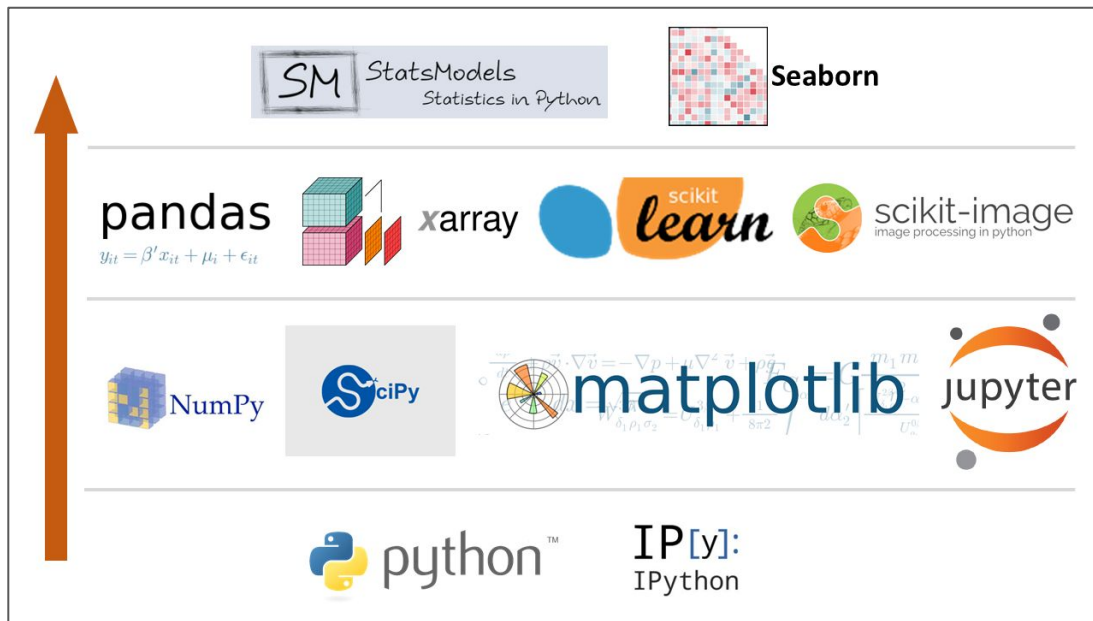
1.1 Giới thiệu Pandas

❖ Pandas là gì?



1.1 Giới thiệu Pandas

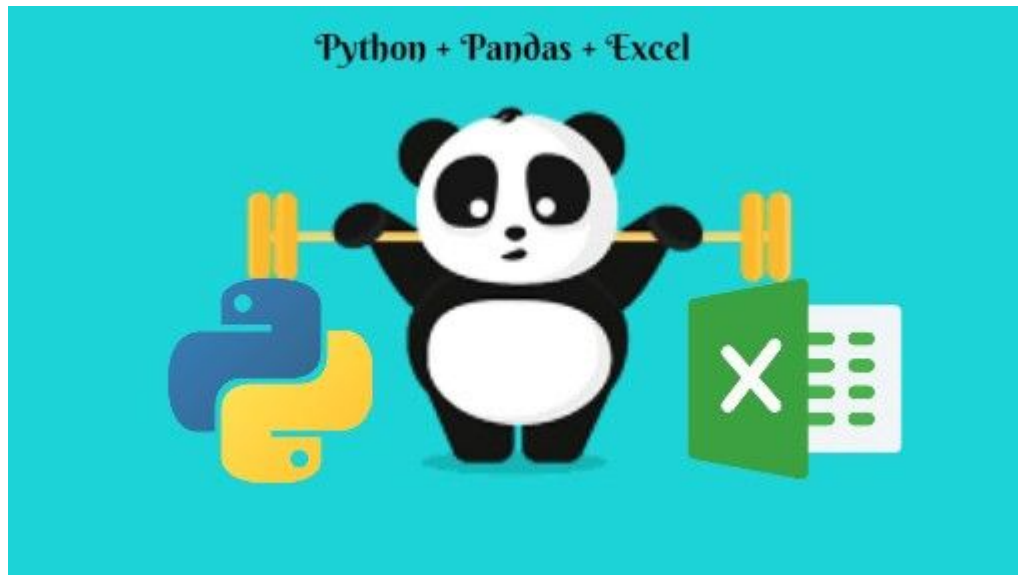
- ❖ Pandas là một trong các thư viện quan trọng nhất trong Python
- ❖ Được phát triển để xử lý dữ liệu có cấu trúc (dạng bảng) và time series



Nguồn: https://cocalc.com/share/public_paths/741ad81231a9db8d8f83bf312458c606ddae7b1d

1.1 Giới thiệu Pandas

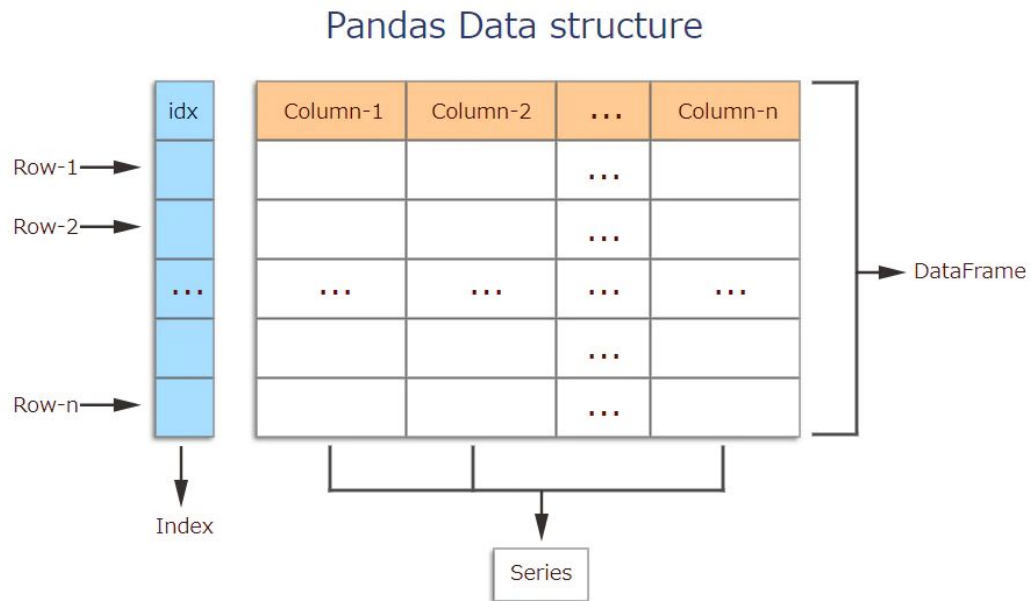
- ❖ Pandas hướng tới việc xử lý dữ liệu dạng bảng tương tự Excel



1.1 Giới thiệu Pandas

Hai loại mảng dữ liệu chính trong Pandas

- ❖ Series: <https://pandas.pydata.org/docs/reference/api/pandas.Series.html>
- ❖ DataFrame: <https://pandas.pydata.org/docs/reference/frame.html>



1.1 Giới thiệu Pandas

Hai loại mảng dữ liệu chính trong Pandas

- ❖ Series: mảng (ndarray) một chiều có nhãn chỉ số
- ❖ DataFrame: mảng hai chiều có nhãn chỉ số
(thường là dạng bảng dữ liệu phức tạp, không đồng nhất)

Series 1			Series 2			Series 3			DataFrame			
Mango			Apple			Banana				Mango	Apple	Banana
0	4		0	5		0	2		0	4	5	2
1	5		1	4		1	3		1	5	4	3
2	6	+	2	3	+	2	5	=	2	6	3	5
3	3		3	0		3	2		3	3	0	2
4	1		4	2		4	7		4	1	2	7

1.1 Giới thiệu Pandas

❖ 10 "amazing applications"

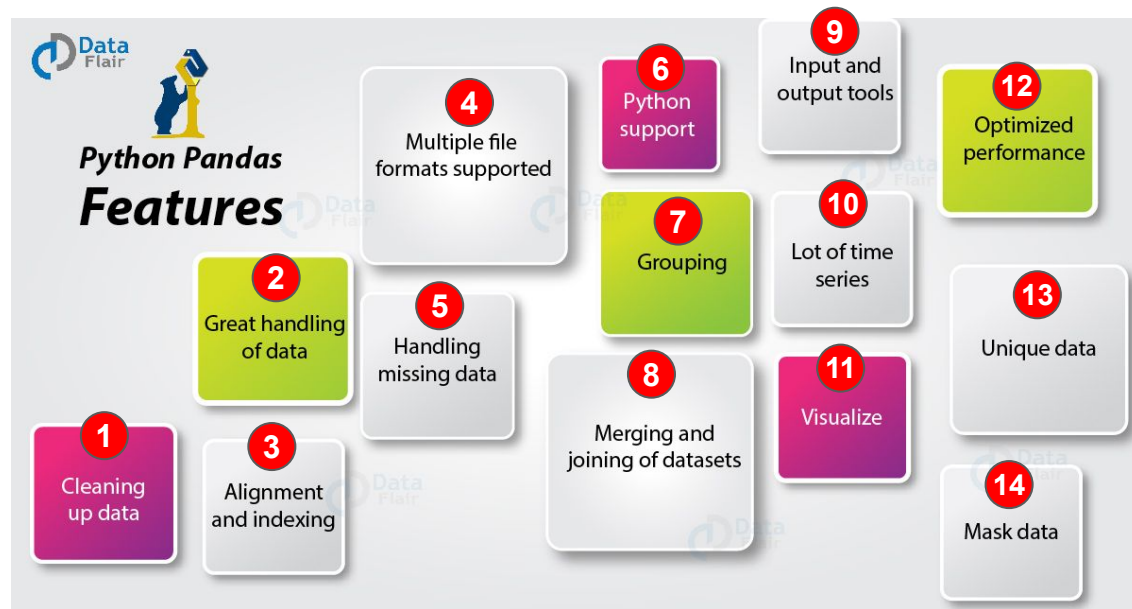
(nguồn: <https://data-flair.training/blogs/applications-of-pandas/>)



1.1 Giới thiệu Pandas

Các tính năng chính của Pandas:

1. Làm sạch dữ liệu
2. Xử lý và khai thác
3. Căn chỉnh và index
4. Nhiều định dạng files
5. Xử lý dữ liệu khuyết
6. Một phần của Python
7. Góm nhóm dữ liệu
8. Trộn/ghép các dataset
9. Đọc/ghi dữ liệu thuận tiện
10. Xử lý dữ liệu time series
11. Trực quan hóa
12. Tối ưu hiệu suất
13. Tìm các giá trị duy nhất
14. Lọc dữ liệu với mask
15. Các phép toán học



Nguồn: <https://data-flair.training/blogs/python-pandas-features/>

1.2 Series

- ❖ Là mảng (ndarray) một chiều có nhãn chỉ số
- ❖ Thông tin chính: Index (s.index) và Data (s.values)
- ❖ Các thông tin khác: dtype, name, copy

```
pd.Series([-2.8, 3, -4.44, 5])
```

↓
abs()

		Data
Index	0	2.80
	1	3.00
	2	4.44
	3	5.00

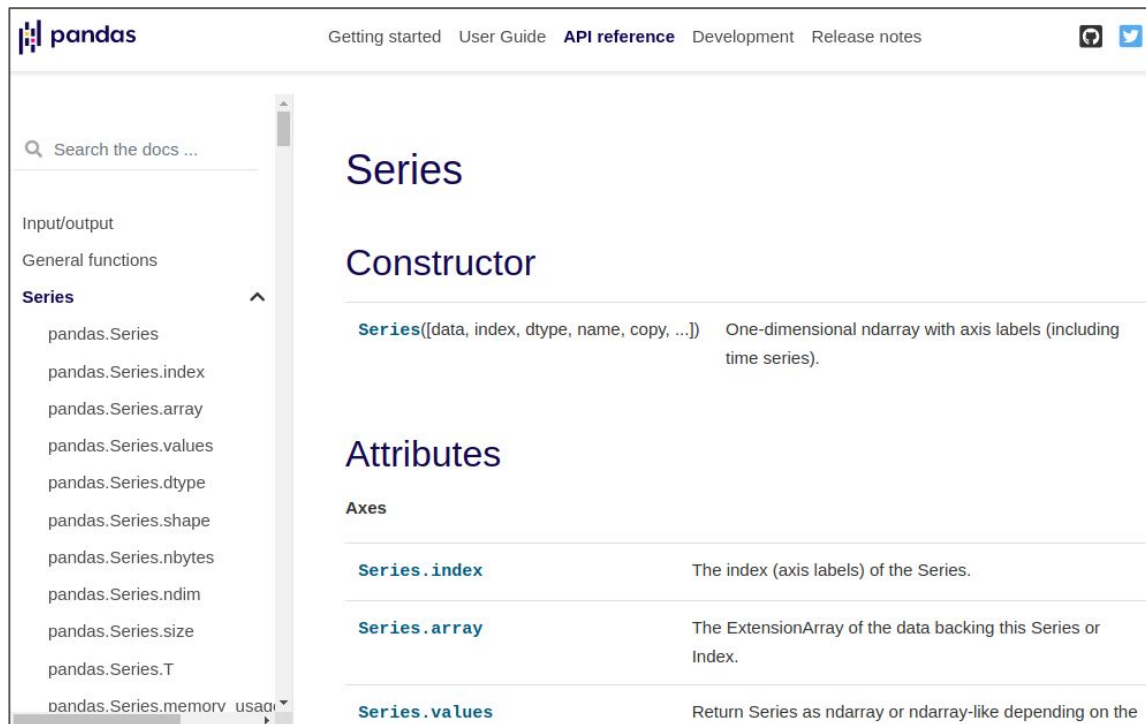


dtype: float64

©v3resource.com

- ❖ Tham khảo tất cả các phương thức trên Series:

<https://pandas.pydata.org/docs/reference/series.html>



The screenshot shows the pandas API reference page for the Series object. The page has a navigation bar with links: Getting started, User Guide, API reference (active), Development, and Release notes. A search bar is present on the left. The left sidebar contains a list of categories: Input/output, General functions, and Series (selected). Under Series, several attributes are listed: pandas.Series, pandas.Series.index, pandas.Series.array, pandas.Series.values, pandas.Series.dtype, pandas.Series.shape, pandas.Series.nbytes, pandas.Series.ndim, pandas.Series.size, pandas.Series.T, and pandas.Series.memory_usage. The main content area is titled 'Series' and 'Constructor'. The constructor is defined as `Series([data, index, dtype, name, copy, ...])` with the description 'One-dimensional ndarray with axis labels (including time series)'. Below this is the 'Attributes' section, which includes a table of attributes:

Axes	
<code>Series.index</code>	The index (axis labels) of the Series.
<code>Series.array</code>	The ExtensionArray of the data backing this Series or Index.
<code>Series.values</code>	Return Series as ndarray or ndarray-like depending on the

Một số thao tác cơ bản trên Series:

- ❖ Slicing:
 - Dựa vào chỉ số: tương tự Numpy
 - Dựa vào hàm:
 - iloc, iat (index dạng số),
 - loc, at (index dạng text)
- ❖ Hiển thị các thông tin cơ bản
 - head(), tail()
 - keys, describe()
- ❖ Reset index: thường thực hiện sau khi chỉ số bị thay đổi do các thao tác

```
new_df = new_df.reset_index(drop=True)
new_df.reset_index(drop=True, inplace=True)
```

1.3 DataFrame

- ❖ Là mảng hai chiều có nhãn chỉ số. Có thể chứa:
 - Các loại dữ liệu khác nhau
 - Dữ liệu khuyết
- ❖ Có hai thông tin chính: Index và Column

	Column names								
	Name	Team	Number	Position	Age	Height	Weight	College	Salary
0	Avery Bradley	Boston Celtics	0.0	PG	25.0	6-2	180.0	Texas	7730337.0
1	John Holland	Boston Celtics	30.0	SG	27.0	6-5	205.0	Boston University	NaN
2	Jonas Jerebko	Boston Celtics	8.0	PF	29.0	6-10	231.0	NaN	5000000.0
3	Jordan Mickey	Boston Celtics	NaN	PF	21.0	6-8	235.0	LSU	1170960.0
4	Terry Rozier	Boston Celtics	12.0	PG	22.0	6-2	190.0	Louisville	1824360.0
5	Jared Sullinger	Boston Celtics	7.0	C	NaN	6-9	260.0	Ohio State	2569260.0
6	Evan Turner	Boston Celtics	11.0	SG	27.0	6-7	220.0	Ohio State	3425510.0

Columns axis=1

Index label

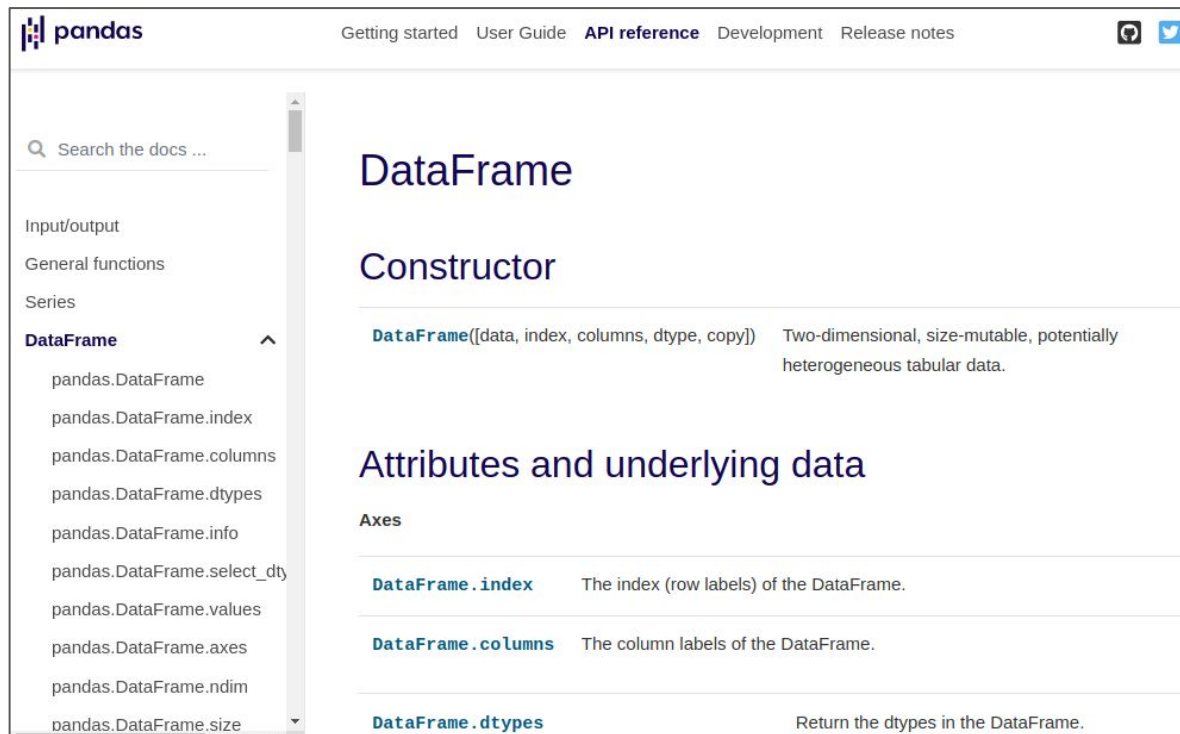
Index axis=0

Missing value

Data

- ❖ Tham khảo tất cả các phương thức trên DataFrame:

<https://pandas.pydata.org/docs/reference/frame.html>



The screenshot shows the pandas documentation website. The top navigation bar includes links for 'Getting started', 'User Guide', 'API reference' (which is active), 'Development', and 'Release notes'. On the left, there is a search bar and a sidebar menu with categories like 'Input/output', 'General functions', 'Series', and 'DataFrame'. The 'DataFrame' category is expanded, showing a list of sub-items including 'pandas.DataFrame', 'pandas.DataFrame.index', 'pandas.DataFrame.columns', 'pandas.DataFrame.dtypes', 'pandas.DataFrame.info', 'pandas.DataFrame.select_dty', 'pandas.DataFrame.values', 'pandas.DataFrame.axes', 'pandas.DataFrame.ndim', and 'pandas.DataFrame.size'. The main content area is titled 'DataFrame' and 'Constructor'. It defines DataFrame as 'Two-dimensional, size-mutable, potentially heterogeneous tabular data.' and shows the constructor signature: `DataFrame([data, index, columns, dtype, copy])`. Below this, the section 'Attributes and underlying data' is shown, with a sub-section 'Axes'. It lists `DataFrame.index` as 'The index (row labels) of the DataFrame.' and `DataFrame.columns` as 'The column labels of the DataFrame.'. At the bottom, it shows `DataFrame.dtypes` as 'Return the dtypes in the DataFrame.'

1.3 DataFrame

❖ Truy cập vào từng cột

```
df['tên-cột']  
df.tên-cột
```

❖ Truy cập vào nhiều cột

```
df[['tên-cột-1', 'tên-cột-4', 'tên-cột-3']]  
# Danh sách tên các cột  
cols = list(df.columns.values)
```

❖ Lọc theo giá trị của cột

```
df.loc[df['tên-cột'] == giá-trị]
df.loc[(df['cột-1'] == gt1) & /| (df['cột-2'] == gt2)]
# Ví dụ: lọc theo chuỗi con 'abc'
df.loc[df['tên-cột'].str.contains('abc')]
```

1.3 DataFrame

- ❖ Sắp xếp theo giá trị của một cột

```
df.sort_values('tên-cột', ascending=True/False)
```

❖ Lặp theo cột: items()

```
# Lấy tên từng cột  
for col in df:  
    ...
```

```
# Lấy dữ liệu từng cột  
for col in df.items():  
    ...
```

❖ Lặp theo dòng: iterrows(), itertuples()

```
for index, row in df.iterrows():  
    ...
```

```
for t in df.itertuples():  
    ...
```

1.3 DataFrame

❖ Tạo cột mới: bằng phép gán

```
# df['tên-cột-mới'] = mảng-giá-trị  
# Ví dụ:  
df['Tổng'] = df.iloc[:, 3:5].sum(axis=1)
```

❖ Thêm cột mới: bằng phép insert

```
# DataFrame.insert(loc, column, value,  
allow_duplicates=False)  
df.insert(1, 'tên-cột-mới', [1, 2, 4, 8])
```

❖ Bỏ một cột

```
df = df.drop(columns='tên-cột')
```

❖ Tự động thiết lập lại index

```
# DataFrame.reset_index(level=None, drop=False,  
inplace=False, col_level=0, col_fill='')  
df.reset_index()
```

❖ Thiết lập cột sẽ trở thành index

```
# DataFrame.set_index(keys, drop=True, append=False,  
inplace=False, verify_integrity=False)  
df.set_index('tên-cột')
```

```
# Thiết lập nhiều cột làm index  
df.set_index(['tên-cột-1', 'tên-cột-2'])
```

1.4 Sự tương quan

- ❖ Giữa từng cặp các cột trong một DataFrame:

```
# DataFrame.corr(method='pearson', min_periods=1)
# method: {'pearson', 'kendall', 'spearman'} or func
df.corr()
```

- ❖ Giữa một DataFrame với một DataFrame/Series khác

```
# DataFrame.corrwith(other, axis=0, drop=False,
method='pearson')
df.corrwith(df2)
```




Phần 2.

Ghi/Đọc files với Pandas

2.1. Ghi/Đọc file csv

2.2. Ghi/Đọc file Excel

❖ Ghi một DataFrame vào file csv:

Tham khảo: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.to_csv.html

```
# DataFrame.to_csv(path_or_buf=None, sep=',', columns=None,
header=True, index=True, index_label=None, mode='w', encoding=None,
compression='infer')
df.to_csv('out.zip', index=False, compression='zip')
# compression: {'infer', 'gzip', 'bz2', 'zip', 'xz', None}
```

❖ Đọc một file csv vào DataFrame

Tham khảo: https://pandas.pydata.org/docs/reference/api/pandas.read_csv.html

```
# pd.read_csv(filepath_or_buffer, sep=',', header='infer',
index_col=None, usecols=None)
df = pd.read_csv('data.csv', index_col=0)
```

2.2 Ghi/Đọc file Excel

❖ Ghi một DataFrame vào file Excel:

Tham khảo: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.to_excel.html

```
# DataFrame.to_excel(excel_writer, sheet_name='Sheet1', na_rep='',  
columns=None, header=True, index=True, index_label=None, startrow=0,  
startcol=0, merge_cells=True, encoding=None)  
df.to_excel("output.xlsx", sheet_name='Sheet_name_1')
```

❖ Đọc một file Excel vào DataFrame

Tham khảo: https://pandas.pydata.org/docs/reference/api/pandas.read_excel.html

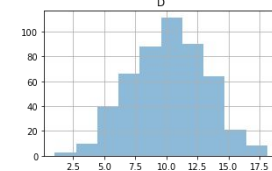
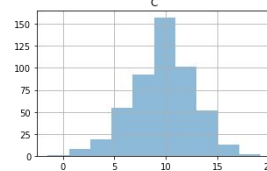
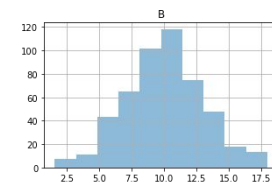
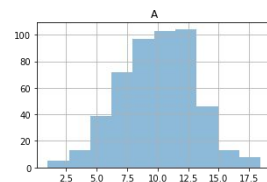
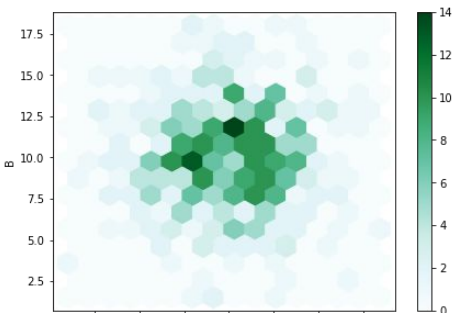
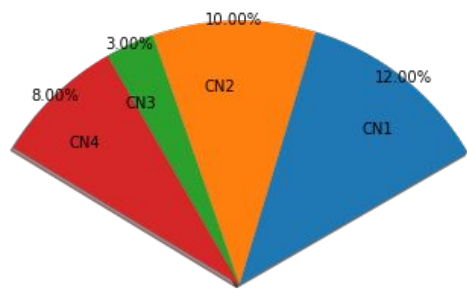
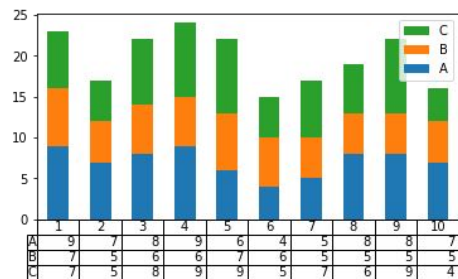
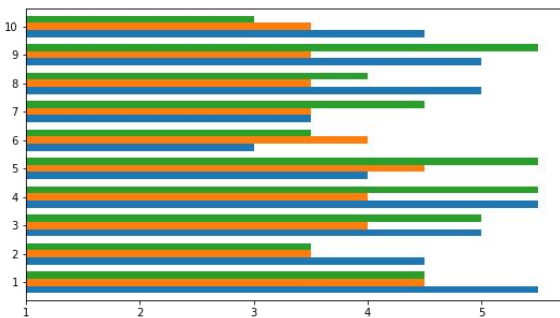
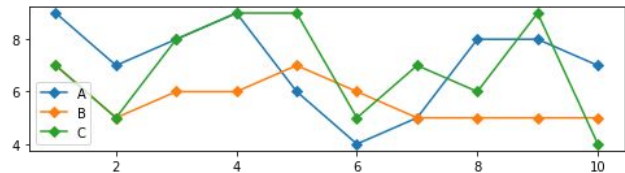
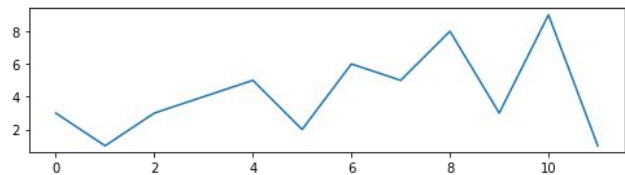
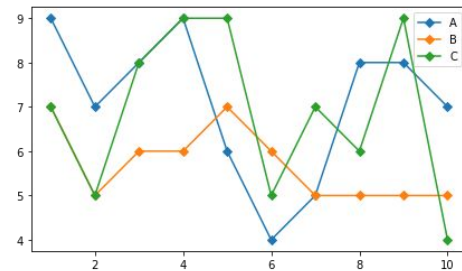
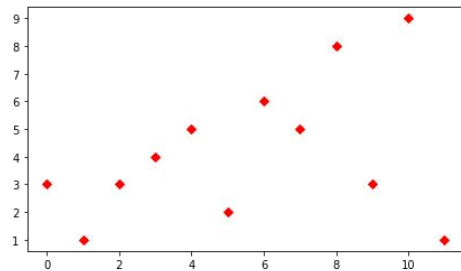
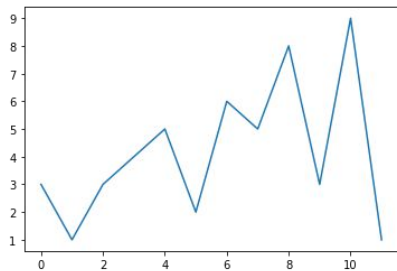
```
# pd.read_excel(io, sheet_name=0, header=0, names=None, index_col=  
None, usecols=None, dtype=None)  
df = pd.read_excel('output.xlsx', index_col=None, header=None)
```



Phần 3.

Vẽ biểu đồ trong Pandas

- 2.1. Biểu đồ điểm và đường
- 2.2. Biểu đồ cột và histogram
- 2.3. Biểu đồ scatter và hexagonal-bin
- 2.4. Biểu đồ tròn và quạt
- 2.5. Hiển thị table trên biểu đồ



Vẽ biểu đồ trong Pandas

Có 2 cách để vẽ:

❖ Dùng hàm `plot()`:

```
df.plot(kind = 'line', ... )
```

- 'line' : line plot (default)
- 'bar' : vertical bar plot
- 'barh' : horizontal bar plot
- 'hist' : histogram
- 'box' : boxplot
- 'kde' : Kernel Density Estimation plot
- 'density' : same as 'kde'
- 'area' : area plot
- 'pie' : pie plot
- 'scatter' : scatter plot (DataFrame only)
- 'hexbin' : hexbin plot (DataFrame only)

❖ Dùng các hàm từ class `plot`:

```
df.plot.bar(), df.plot.barh()
```

```
df.plot.hist()
```

```
df.plot.scatter(), df.plot.hexbin()
```

20 hàm Pandas "must know" cho EDA

1. df.head()	Trả về các dòng đầu của mảng (default: 5 dòng)
2. df.tail()	Trả về các dòng cuối của mảng (default: 5 dòng)
3. df.info()	Tóm tắt nhanh về một DataFrame
4. df.shape	Hình dạng
5. df.size	Số các phần tử
6. df.ndim	Số chiều
7. df.describe()	Tóm tắt thống kê của một DataFrame
8. df.sample()	Tạo ra một random sample theo dòng hoặc cột
9. df.isnull().sum()	Kiểm tra các giá trị khuyết (missing values)
10. df.nunique()	Số các phần tử duy nhất trong một DataFrame

Tham khảo: <https://www.analyticsvidhya.com/blog/2021/04/20-must-known-pandas-function-for-exploratory-data-analysis-eda/>

20 hàm Pandas "must know" cho EDA (tt.)

11. df.index	Mảng index
12. df.columns	Mảng các columns
13. df.memory_usage()	Tóm tắt về chiếm dụng bộ nhớ của mảng
14. df.dropna()	Bỏ đi các dòng NaN
15. df.nlargest()/df.nsmallest()	Trả về n dòng có giá trị lớn nhất/nhỏ nhất sắp theo 1 cột nào đó
16. df.isna()	Kiểm tra phần tử NaN
17. df.duplicated()	Trả về các dòng giống hệt nhau
18. value_counts()	Đếm số phần tử duy nhất
19. df.corr()	Tính hệ số correlation coefficient (R)
20. df.dtypes	Kiểu dữ liệu

Tham khảo: <https://www.analyticsvidhya.com/blog/2021/04/20-must-known-pandas-function-for-exploratory-data-analysis-eda/>

THANK YOU!

